COMP1521 – Computer Systems Fundamentals

Created: 22 Apr 2015 Proposal Last Updated: 28 Apr 2016

Offering Details:

Key Details and Contacts

Key Course Details	
Course Name (Official)	Computer Systems Fundamentals
Standard Name (SIMS)	Computer Systems Fundamentals
Course Code	COMP1521
Units of Credit (UOC)	6
Career	Undergraduate
Level	1
First Semester and Year the Course will be offered	2017 Semester 1
Does this new course replace another existing course?	Yes
Courses to be replaced	Not specified

Contact Details

	Name	Email	Role
Proposal Proponent	John Shepherd	jas@cse.unsw.edu.au	Senior Lecturer, School of Computer Science and Engineering
Proposal Author(s)	Not specified		
Proposal Contact	John Shepherd	jas@cse.unsw.edu.au	Senior Lecturer, School of Computer Science and Engineering
	Kerstin Vintila	z3028233@unsw.edu.au	Administrative Officer, School of Computer Science and Engineering
Optional Additional Endorsers	Not specified		
Academic Unit responsible for course	School of Computer Science and Engineering		
Parent Academic Unit	Faculty of Engineering		

Proposal Concept

Summary of Proposal

Summary of Proposal

This course gives an introduction to the study of computer systems.

The course is part of CSE's core syllabus redevelopment. It is intended to be the first course in computer systems taken by all students enrolled in a CSE-run degree or any dual-award program that includes a CSE-run program. It does not replace any existing courses, although it has some overlap with the existing COMP2121, which will be revised.

This course is not intended to be exclusive to CSE students. Students from other disciplines are welcome to enrol.

Justification for proposal

Justification for Proposal

As part of on-going curriculum development, CSE has redesigned its lower-level courses based on the ACM Computing Curriculum, to include all of the topics that we believe every CSE graduate should know about when they graduate. This has led to the development of five new courses, which effectively replace the existing COMP1917, COMP1927 and COMP2911 courses. In addition, SENG1031 (the first software engineering workshop) will be replaced by an introductory software engineering course that all students in CSE degrees must take. Also, the other level-2 courses COMP2121 and COMP2041 will be re-designed to complement the new core syllabus.

Note that while these courses are targeted at students in CSE degrees, we encourage students from other programs to enrol if they want a more comprehensive introduction to computing than what is available in the service courses COMP1911, COMP1921 and ENGG1811. These service courses will, however, be retained for the majority of Engineering students who do not think they need such a detailed view of computing.

Anticipated Enrolments

Anticipated Enrolments for next 3 years

2017	2018	2019
400	400	400

Attachments	
Attach documentation to this proposal	None attached

Learning and Teaching

Learning & Teaching development and support	
Are there Learning & Teaching space requirements for the course beyond those that can be accommodated by CATS spaces?	No
Have you discussed with the Learning Centre and Learning and Teaching what language and/or academic skills development resources and/or which teaching and learning strategies might be suited to this course?	No
Are many students in this course at a key transition point where their academic skills are likely to need development, e.g. from one kind of educational institution or type of program to another or into education after a significant break?	No

Consultation

Internal consultation

Internal Consultation	**Consultants**	None specified
	Details	Jingling Xue ran the Working Group which developed the new core syllabus. The Working Group included academics across a range of computing disciplines and levels.
	Attachments	None specified

External consultation

External Consultation	**Consultants**	None specified
	Details	Other Engineering schools whose students we believe might be interested in COMP1521 (specifically, EET and Mechatronics students) were consulted on the syllabus and have indicated that they are happy with the content.
	Attachments	None specified
Interested Parties	Not specified	

Related Proposals

Related Proposals

Code	Proposal Name	Type	Date	Status
COMP1511	Introduction to Programming	New Course (UG)	Apr 2015	Submitted
COMP1531	Software Engineering Fundamentals	New Course (UG)	Apr 2015	Draft Proposal

Endorsements and Comments

Endorsement history	No endorsements have been recorded for this proposal (yet).
Comments	No comments posted

Administration:

Key Course Details

Key Admin Details	
Course Name (Official)	Computer Systems Fundamentals
Student System ID	A Student System ID will be generated once this course is approved.
Can course be taken as General Education elective?	Yes
Field of Education	031305 – Computer Engineering

Course Review	
Next course review date	December 01, 2018
Provide details of any particular factors that need to be considered at that review.	The course will be reviewed after each offering in 2017 to check how effectively it is meeting its objectives. A formal review will be conducted at the end of 2018.

Delivery and Attendance

Campus administering the Course: Sydney

Teaching Shares by School/Faculty

School	Teaching Share (%)
School of Computer Science and Engineering	100
Total Share	**100**

Semesters the course is offered

	Summer Semester	Semester 1	Semester 2
2015	No	No	No
2016	No	**Yes**	**Yes**
2017	No	**Yes**	**Yes**
2018	No	**Yes**	**Yes**
2019	No	**Yes**	**Yes**

Teaching mode and contact hours: Standard Offering Mode

Standard offering contact hours per week

Learning Activity	Hours/Week
Lecture	3
Tutorial/Laboratory	0
Tutorial	1
Laboratory	2
Web-based Online Learning Activity	0
Clinical/Fieldwork	0
Distance Learning	0
Seminar	0
Studio	0
Meeting/Consultation	0
Total Hours per week	**6**

Primary delivery mode: Classroom

Secondary delivery modes: Online

Additional information about the delivery modes for this course: All course materials will be available online. Students must attend tutorials/laboratories. Students should attend lectures.

Staff

Staff associated with course

Course Convenor

Name	Email	Role
Hui Guo	z3068443@unsw.edu.au	Senior Lecturer, School of Computer Science and Engineering
John Shepherd	jas@cse.unsw.edu.au	Senior Lecturer, School of Computer Science and Engineering

Administrative Contact	Name	Email	Role
	Kerstin Vintila	z3028233@unsw.edu.au	Administrative Officer, School of Computer Science and Engineering

Supplementary Information:

Resources

Student Resources	
Prescribed Resources	None specified
Recommended Resources	None specified

Experience and Assumed Knowledge

Industrial Experience Component	
Industrial Experience Component	None

Assumed Knowledge	
Assumed Knowledge	We assume that students have completed (at least) an introductory programming course. We assume no prior knowledge of computer systems.

Academic Structure:

Academic Structure

Prerequisites	
Prerequisite courses	COMP1511 - Introduction to Programming (UG)
Prerequisite programs	Not specified
Prerequisite streams	Not specified
Prerequisite conditions	Not specified

Exclusions	
Excluded Courses	Not specified
Excluded Programs	Not specified
Excluded Streams	Not specified

Equivalent	
Equivalent courses	Not specified

Assessment

Assessment	
Grading Basis	Standard UNSW grades (e.g. HD, DN, CR, PS, FL)

Assessment items and their relationship to Course Learning Outcomes

	Assessment Title	Assessment Type	Weight (%)
1	Final Exam	Examination	65%
	Assessment Description:	The final examination will include both Practical and Written components. Feedback via final mark.	
2	Machine-level Programming	Assignment	10%
	Assessment Description:	Not specified	
3	Memory Hierarchies	Assignment	10%
	Assessment Description:	Not specified	
4	Parallelism	Assignment	10%
	Assessment Description:	Not specified	
5	Systems Laboratory	Lab Work	5%
	Assessment Description:	Not specified	
Total Weight			**100%**

Final Exam

- describe the layers of architectures in modern computer systems from hardware device levels upwards
- describe the principles of memory management and explain the workings of a system with virtual memory management
- explain how the major components of a CPU work together, including how data (including instructions) is represented in a computer
- design, implement and analyse small programs at the assembly/machine level, including the use of I/O, interrupts and traps
- describe the relationship between high-level procedural languages (e.g., C) and assembly/machine language in the conventional machine layer, including how a compiled program is executed in a classical von Neumann machine, with extensions for threads, multiprocessor synchronization, and SIMD execution
- explain how input/output operations are implemented, and describe some basic I/O devices
- describe the layered structure of a typical networked architecture

Machine-level Programming

Memory Hierarchies

Parallelism

Systems Laboratory

Curriculum Mapping

Course Learning Outcomes

Specify the learning outcomes that students should achieve upon successful completion of this course

1	describe the layers of architectures in modern computer systems from hardware device levels upwards
2	describe the principles of memory management and explain the workings of a system with virtual memory management
3	explain how the major components of a CPU work together, including how data (including instructions) is represented in a computer
4	design, implement and analyse small programs at the assembly/machine level, including the use of I/O, interrupts and traps
5	describe the relationship between high-level procedural languages (e.g., C) and assembly/machine language in the conventional machine layer, including how a compiled program is executed in a classical von Neumann machine, with extensions for threads, multiprocessor synchronization, and SIMD execution
6	explain how input/output operations are implemented, and describe some basic I/O devices
7	describe the layered structure of a typical networked architecture

Teaching strategies and Rationale

Teaching Strategies and Rationale	Lectures, laboratory exercises, take-home programming assignments

Course Aims	
Course Aims	• Provides a programmer's view on how a computer system executes programs, manipulates data and communicates • Enables students to become effective programmers in dealing with issues of performance, portability, and robustness • Serves as a foundation for later courses on networks, operating systems, computer architecture and compilers, where a deeper understanding of systems-level issues is required

Publications and Marketing:

Publications

Course Description	
Description of course that can be used in online publications (e.g. Handbook website, Faculty websites or other online catalogue systems)	Introduction to the systems-level view of computing. Topics: number representation, machine-level programming, representing high-level programs in machine code, memory, input/output, system architectures, parallelism/concurrency, communication/synchronisation.

Key Search Terms	
List key search terms that might be used to search for this course (e.g. via the Handbook or Google searches).	programming computing systems